InnoVote CardReader

Functional Design

By
Erin Thead
Software Engineer
erin@erinthead.com

Table of Contents – CardReader Functional Design

1. Introduction

1.1. Purpose.

The purpose of this document is to communicate the software requirements and functional design for the InnoVote CardReader Software. The document provides a detailed description of functional, performance, and security requirements, design constraints, and classes of persons who will be using the software.

The intended audience of this document is the developer and any other persons interested in the project, including election reform activists, computer security professionals, political figures with an interest in election reform, and potential buyers of the design.

1.2. Scope.

The InnoVote CardReader Software is one component of an interoperable line of products. It is a software product designed to execute on a computer connected to the proposed "CardReader" ballot-scanning machine [ref. 5].

CardReader Software will handle input from the CardReader hardware components, and, when necessary, send output to them. CardReader Software will manipulate data received from the CardReader hardware as well as from other InnoVote products as necessary.

This document does not address hardware-operation and hardware-management issues except as they relate to a software operation related to conducting an election. Reference [1], *CardReader Hardware Requirements Overview*, addresses such low-level hardware operation details.

Current ballot-scanning technology is demonstrably insecure and unreliable. Existing DRE machines, which are used in voting sites across the United States, have had numerous and severe documented errors, including the following:

- A base error rate of 1-2%
- Votes being counted for the wrong candidate
- The suspicion of pre-election rigging of machines
- Unverified updates being made to election software
- Public online access to sensitive operations of election software

In contrast to most current voting software, CardReader is a highly secure and accountable system with strong anti-fraud protection. Election data are heavily protected both from errors and from tampering.

1.3. References.

[1] Thead, E. *InnoVote CardReader Hardware Requirements Overview*, 2005.
[2] Thead, E. *InnoVote Database Detailed Design*, 2005.
[3] Thead, E. *InnoVote MyVotronic Hardware and Operating System Overview*, 2005.
[4] Thead, E. *InnoVote Network Detailed Design*, 2005.
[5] Thead, E. *InnoVote ReliaVote Central Server Functional Design*, 2005.
[6] Thead, E. *InnoVote ReliaVote Precinct Edition Functional Design*, 2005.
[7] Thead, E. *InnoVote SecureDRE Functional Design*, 2005.
[8] Thead, E. *InnoVote Database Access Matrix*, 2005.
[9] Thead, E. *Security Analysis of InnoVote Products*, 2005.

1.4. Overview.

The remainder of this document is organized in the following fashion:

Section 2: Provides overall description of the System, including product functions, user classes, assumptions, and generalized dependencies.

Section 3: Provides specific requirements including functional requirements, performance requirements, and security requirements.

Section 4: Provides supporting figures and tables for information in other sections of the document.

2. Overall Description

2.1. Product Functions.

CardReader Software will need to perform the following basic functions:

1. Accept data transfers from the ReliaVote Precinct Edition software product "programming" the machine. This involves adding and/or modifying entries in Database tables.
2. Allow a user to program the Hardware's scanner to correctly read votes from a paper ballot.
3. Scan a voter's paper ballot and record his/her choices in the Database.
4. Finalize a vote and lock it from further changes
5. Transmit votes in real-time to the ReliaVote Precinct Edition software
6. Accept a signal from ReliaVote Precinct Edition indicating that the election is ended
7. Lock input from and output to hardware when an election is over
8. Unlock hardware to allow input and output
9. Send the final vote results and tallies to the precinct computer
10. Conduct recounts and send the results to the precinct computer
11. Detect and log error conditions
12. Remove old vote data from the System when the data are no longer needed

2.2. User Classes.

2.2.1. Voter user class

The Voter user class represents a citizen of the United States who is casting a ballot in a primary or general election. This class of user will need to perform operations that modify sensitive tables in the InnoVote Databases.

The Voter user class is assigned to all input from the CardReader Hardware until the Software assigns it a different privilege level.

Members of the Voter class must be allowed to perform operations that will add entries to the Votes and Realtime_Votes tables in the Database and modify the Tallies table. It is *not* necessary for these operations to be performed at the Voter privilege level. The operations can be initiated by a Voter and the changes to the Database can be performed by the System user.

2.2.2. Election Official user class

The Election Official user class represents a citizen of the United States who has the legal authority to oversee a general or primary election in a precinct. This class of user will need to initiate operations that modify sensitive tables in the InnoVote Database but should *not* have direct access to the Database.

This user class is assigned to any incoming data transfers that originated from ReliaVote PE until the Software assigns a different privilege level. It is assumed that only election officials or persons acting under the legal authority of election officials will be using the ReliaVote software.

The Election Official does not have direct access to the Database. Any Election Official requests for modifications to the database will be performed at the System privilege level.

2.2.3. System user

The Software shall function as a user class. It must be able to perform any operation necessary on the hardware and data stored in the System, within the permissions of the computer's operating system. Operations in software functions initiated by Voters or Election Officials may be executed at System level. These functions are described in §3 of this document.

The System will in theory have full access to the Database at all times, but the software must be coded so that it will in *practice* not perform certain damaging operations to the Database, and that no human user or external machine will be able to interact with SecureDRE with System privileges. Additionally, the Database shall be configured to disallow certain actions [reference 2].

2.3. Assumptions and Dependencies.

2.3.1. Hardware platform assumption

The document assumes that InnoVote CardReader Software will operate on a computer permanently connected to the ballot-scanning hardware "CardReader" or compatible hardware. CardReader Software has numerous functions that require certain hardware features for correct operation.

2.3.2. Internet connectivity assumption

The document assumes that hardware running InnoVote CardReader will be connected to a TCP/IP-compatible network during operation. IP version 6 is not necessary for correct operation of CardReader, but it does provide extra security benefits.

2.3.3. Single-user assumption

The document assumes that CardReader is a single-user software product. Although there is no provision for user profiles and logins, it is conceivable that CardReader could receive instructions from more than one concurrent source or "user," as defined by the User Classes (§2.2) in this document.

For any set of operations that could potentially be in conflict, an operation currently being executed by the System has precedence. Next are operations that were initiated by Election Officials. Lowest in precedence are operations that were initiated by Voters.

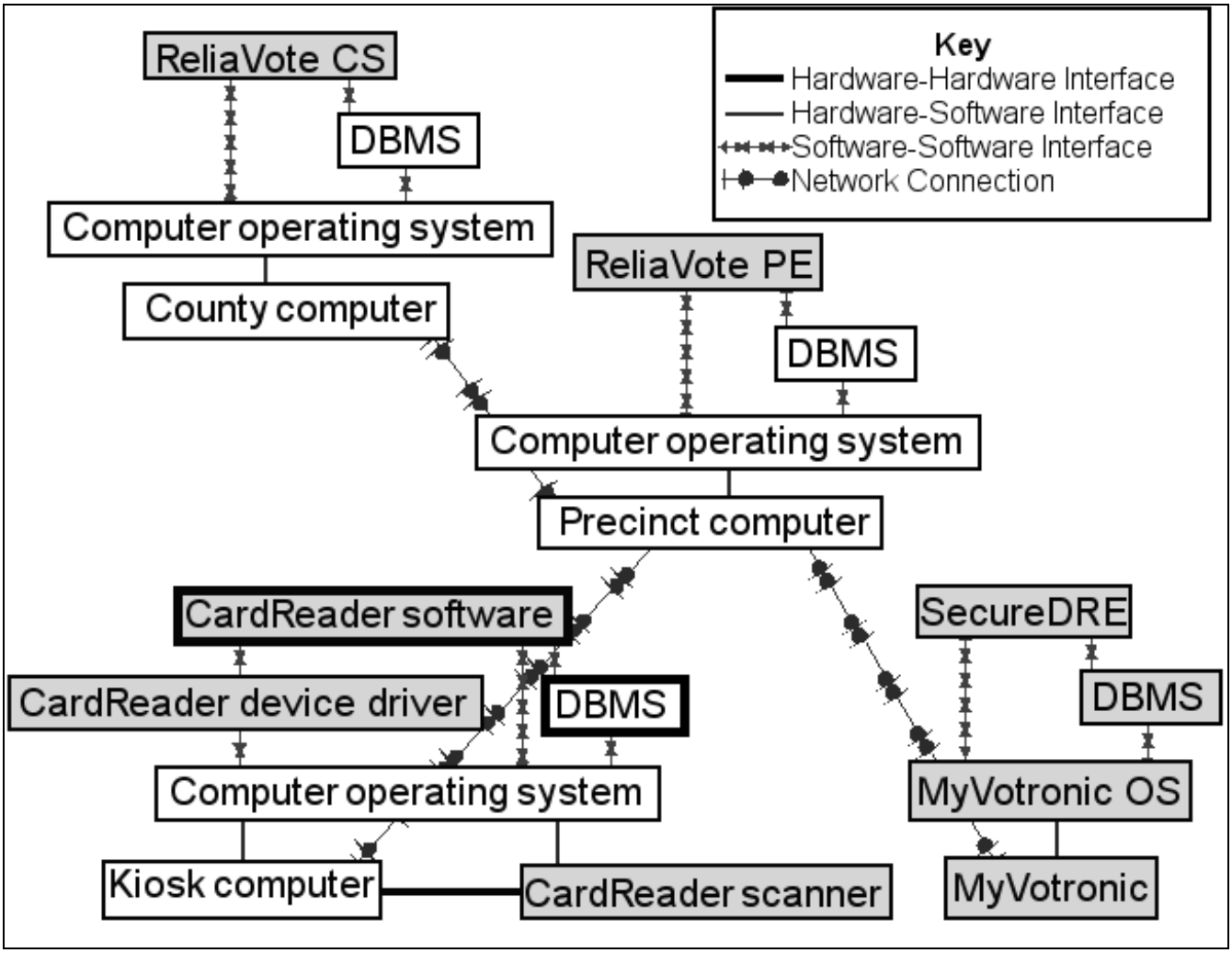
2.3.4. Precinct computer assumption

The document assumes that a CardReader machine running CardReader Software will be connected over a network to one and only one computer that is running "InnoVote ReliaVote Precinct Edition" or compatible software. Several operations of the CardReader product require receiving data from ReliaVote Precinct Edition for correct operation.

2.4. Deployment of the Software.

Figure 1 shows the deployment diagram for all InnoVote products and necessary third-party components. Items that this document describes are surrounded with thick boxes.

Figure 1: Deployment Diagram.



3. Specific Requirements

3.1. Functional Requirements.

The features described in this section are operations that are necessary for correct and useful operation of the CardReader Software product.

3.1.1. System Feature 1: Enter private network

3.1.1.1. Purpose of Feature

This feature allows the Software to establish a unique IP address on a precinct-level private network, open a data-transfer port for CardReader Software to use, and identify itself to the precinct computer. The precinct computer can then maintain a table of all devices that are located on the network.

3.1.1.2. Stimulus-Response Sequence

1. The CardReader Software scans through the CardReader kiosk's 65,535 data transfer ports and determines whether any of them are currently open other than a single port that is used for communication with the precinct Kerberos server. Conditional: If any other port is open by another application, the software creates an entry in the Events log of the Database containing the name of the service and the port number in use. When it reaches the last port, the Software pauses execution and displays a message to the user containing any open port numbers and the applications using them, informing the user that it will not continue execution until the ports are closed.
2. CardReader determines a single data-transfer port to use on the network. This port should be the same for all installations of CardReader Software.
3. The System receives a data transmission from the precinct computer, which has recognized the machine's presence on the network.
4. The System accesses the key management server and verifies the data's encryption keys.
5. The System decrypts the data and determines them to be requests from the precinct computer to for its type of equipment, its hardware version, its software version, and its current status.
6. The System transmits the requested information to the precinct computer.
7. The System records an entry in its Events table indicating the event.

3.1.1.3. Dependencies

This operation requires the completion of no other System Features before it can begin execution.

This feature is restricted by Security Feature 7, "Block all ports except two." The feature requires Security Features 1, "Verify identity of data transmitters," and 6, "Encrypt outbound data."

3.1.2. System Feature 2: Program the Database

3.1.2.1. Purpose of Feature

This feature allows the System's Database to be modified so that the appropriate races, candidates, and candidate information are displayed to the Voter. In a traditional context, this is called "programming" the voting equipment.

3.1.2.2. Stimulus-Response Sequence

1. The System receives one or more data transmissions from the precinct computer.
2. The System verifies the data's encryption keys.
3. The System decrypts the data and determines them to be instructions to add entries to the Candidates, Parties, Contests, Precincts, Running, and/or Affiliations tables in its Database.
4. The System attempts to add the specified entries to the correct tables in the Database.
5. The System records an entry in its Events table indicating the success or failure of the operation.
6. The System sends a notification to the precinct computer of its success or failure to add the data to its database. Conditional: If the machine failed to add the data, the notification contains the error code and error text.

3.1.2.3. Dependencies

This operation requires the completion of System Feature 1, "Enter private network," before it can begin execution.

This feature is restricted by Security Features 4, "Limit changes to Database on Election Day," 7, "Block all ports except two," and 8, "Database login." The operation requires Security Features 1, "Verify identity of data transmitters," 2, "Encrypt Database tables," and 6, "Encrypt outbound data."

3.1.3. System Feature 3: Program the scanner

3.1.3.1. Purpose of Feature

This feature allows the System to accept programming from ReliaVote Precinct Edition that programs the ballot scanner to recognize the order of the races and candidates on a paper ballot. The Hardware is then able to categorize the input it receives in System Feature 4, "Scan ballot."

3.1.3.2. Stimulus-Response Sequence

1. The System receives one or more data transmissions from the precinct computer.
2. The System verifies the data's encryption keys.
3. The System decrypts the data and determines them to be instructions to configure the ballot scanner to associate markings on particular areas of a ballot with particular options in the Candidates table of the Database.
4. The System attempts to update the ballot scanner's configuration.
5. The System records an entry in its Events table indicating the success or failure of the operation.
6. The System sends a notification to the precinct computer of its success or failure to update the configuration. Conditional: If the machine failed to configure the scanner, the notification contains the error code and error text

3.1.3.3. Dependencies

This feature requires the successful completion of System Feature 2, "Program the Database," before it can begin execution.

3.1.4. System Feature 4: Scan ballot

3.1.4.1. Purpose of Feature

This feature allows the System to scan a ballot inserted into the scanner by a Voter, detect the Voter's selections, and enter the Voter's selections in the vote Database.

3.1.4.2. Stimulus-Response Sequence

1. A Voter inserts a ballot into the CardReader's ballot scanner.
2. The scanner scans the ballot and detects the Voter's markings.
3. The software converts the output of the bar code scanner to a digital format.
4. The System compares the digitized bar code to the numerical component of the ballot identification tags in the Ballots table of the Database. Conditional: If the machine is not in "Recount Mode" (§3.1.12) and the ballot identification exists, the machine halts execution of these steps and displays a message that the ballot has been read.
5. The System appends an alphabetical code to the front of the ballot's digitized bar code. This alphabetical code identifies the machine through which the ballot was scanned. Conditional: If the machine is in "Recount Mode" and the ballot has been confirmed to have been counted in the initial count, the software bypasses this step.
6. The System compares the Voter's markings to the mappings of ballot areas and Database ballot options.
7. The System generates votes in temporary memory from marked areas of the ballot that are mapped to candidates and ballot options.
8. The System checks each contest for the presence of "overvotes."
   a. If the System detects that a contest has no more than one ballot option selected, it performs Step 7.
   b. If the System detects a vote for more than one candidate for the same contest, it discards all the user's choices for that contest.
9. The System adds the vote in RAM to the Votes and Realtime_Votes Database tables. It then increments the Tallies table entry for the chosen ballot option by one.
10. The System records an entry in its Events table indicating the success or failure of the operation.

3.1.4.3. Dependencies

This feature requires the successful completion of System Feature 3, "Program the scanner," before it can begin execution.

This feature requires Security Feature 2, "Encrypt Database tables." It is restricted by Security Features 3, "Restrict data flow to Database tables," and 8, "Database login."

3.1.5. System Feature 5: Finalize vote

3.1.5.1. Purpose of Feature

This feature locks a vote from modification.

3.1.5.2. Stimulus-Response Sequence

1. The System receives a signal from the precinct computer.
2. The System determines the signal to contain instructions to lock the ballot that was just cast.
3. The System sets the ballot-lock attribute of the last ballot in the Ballots table to “TRUE.”
4. The System records an entry in its Events table indicating the success or failure of the operation.
5. The System sends a transmission to the precinct computer that it has successfully locked the ballot.

3.1.5.3. Dependencies

This feature requires the successful completion of System Feature 4, “Scan ballot,” before it can begin execution.

This feature requires Security Features 1, “Verify identity of data transmitters,” and 6, “Encrypt outbound data.”

3.1.6. System Feature 6: Send real-time election results

3.1.6.1. Purpose of Feature

This feature allows the System to send individual votes to a central tabulation server as they are cast, before the election is ended. The full Votes table is sent when the election is declared over. This duplication of data provides anti-fraud and error-checking protection, since (as described in reference [7]) ReliaVote PE checks the Votes table it receives against the Realtime_Votes table that it has accumulated over the course of the election.

3.1.6.2. Stimulus-Response Sequence

1. The System creates a copy in RAM of the latest-added entry to the "Ballots" Database table, in plain-text. Conditional: If the latest entry is a modification of a ballot for which a paper receipt has already been printed, the System also creates a message in RAM containing the ballot identification for the Voter's previous ballot and a request to the precinct computer to archive that ballot entry.
2. The System creates a copy in temporary memory of the latest-added entry to the "Realtime_Votes" Database table, in plain-text.
3. The System transmits a request to the precinct computer to send the real-time data.
4. The System receives a request from the precinct computer for the data.
5. The System transmits the copies of these table entries to the precinct computer.
6. The System receives an acknowledgment from the precinct computer containing the precinct computer's success or failure to add the data to its own database. Conditional: If the System receives notification of a failure, it records an error entry in its Events table and reverts to Step 3. Conditional: If the System receives a transmission from the precinct computer to lock itself, it records an event in its Events table and immediately performs System Feature 7, "Lock CardReader machine."

3.1.6.3. Dependencies

This feature requires the successful completion of System Feature 5, "Finalize vote," before it can begin execution.

This feature requires Security Features 2, "Encrypt Database tables," 6, "Encrypt outbound data," and 8, "Database login."

3.1.7. System Feature 7: Lock CardReader machine

3.1.7.1. Purpose of Feature

This feature allows for locking of a CardReader machine to block input from a voter and disable all hardware output except for the network adapter.

3.1.7.2. Stimulus-Response Sequence

1. The System receives a signal from the precinct computer.
2. The System determines the signal to contain instructions to lock the machine from accepting input from a voter.
3. The System disables all hardware input and output except from the network adapter.
4. The System records an entry in its Events table indicating the success or failure of the operation.
5. The System sends a data transmission to the precinct computer that it has successfully locked its hardware from accepting input or producing output.

3.1.7.3. Dependencies

This feature requires the successful completion of no other System Features before it can begin execution.

This feature requires Security Features 1, “Verify identity of data transmitters,” and 6, “Encrypt outbound data.”

3.1.8. System Feature 8: Unlock CardReader machine

3.1.8.1. Purpose of Feature

This feature allows CardReader Software to process a request from the precinct computer to unlock the voting machine and allow a voter to interact with it.

3.1.8.2. Stimulus-Response Sequence

1. The System receives a signal from the precinct computer.
2. The System determines the signal to contain instructions to unlock the machine because a new Voter is ready to vote or the election has ended.
3. The System configures the CardReader machine to allow ballot-scanning.
4. The System records an entry in its Events table indicating the success or failure of the operation.
5. The System sends a data transmission to the precinct computer that it has unlocked the hardware and exits this sequence of steps.

3.1.8.3. Dependencies

This feature requires the successful completion of System Feature 7, “Lock CardReader machine,” before it can begin execution.

This feature requires Security Features 1, “Verify identity of data transmitters,” and 6, “Encrypt outbound data.”

3.1.9. System Feature 9: End election

3.1.9.1. Purpose of Feature

This feature allows an Election Official to declare the election over and lock the machine from further voting for a predefined time period.

3.1.9.2. Stimulus-Response Sequence

1. The System receives one or more data transmissions from the precinct computer.
2. The System verifies that the data transmissions actually *are* from the precinct computer.
3. The System decrypts the data and determines them to be instructions to end the election and ignore all input for a given period of time.
4. The System compares its internal time to the time at which the polls are to close. Conditional: If the System's internal time is later than the poll-closing time, it performs Step 5. If the System's internal time is earlier than the poll-closing time, the System discards the data and logs an error in the Events table, because the presence of the erroneous data transmission indicates a problem with the precinct computer.
5. The System locks all unlocked and unarchived Ballots table entries.
6. The System records an entry in its Events table indicating the success or failure of the operation.
7. The System performs System Feature 7, "Lock CardReader machine," and exits this sequence of steps.

3.1.9.3. Dependencies

This feature requires the completion of no other System Features before it can begin execution.

This feature is restricted by Security Features 4, "Limit changes to Database on Election Day," and 7, "Block all ports except two." The feature requires Security Features 1, "Verify identity of data transmitters," and 6, "Encrypt outbound data."

3.1.10. System Feature 10: Send finalized election results

3.1.10.1. Purpose of Feature

This feature allows the System to transmit the final candidate tallies and individual vote Database entries to a central tabulation server.

3.1.10.2. Stimulus-Response Sequence

1. The System creates a copy in RAM of the latest-added entry to the "Ballots" Database table, in plain-text. Conditional: If the latest entry is a modification of a ballot for which a paper receipt has already been printed, the System also creates a copy in RAM of the previous entry in the "Ballots" Database table, which has been archived.
2. The System creates a copy in RAM of the latest-added entry to the "Realtime_Votes" Database table, in plain-text.
3. The System creates a copy in RAM of the "Votes" Database table, in plain-text.
4. The System creates a copy in RAM of the "Tallies" Database table, in plain-text.
5. The System transmits a request to the precinct computer to send the data.
6. The System receives a request from the precinct computer for the data.
7. The System transmits the copies created in Steps 1 – 4 to the precinct computer.
8. The System receives an acknowledgment from the precinct computer containing the precinct computer's success or failure to add the data to its own database. Conditional: If the System receives notification of a failure, it reverts to Step 5. Conditional: If the System receives a transmission from the precinct computer to lock itself, it immediately performs System Feature 7, "Lock CardReader machine."
9. The System records an entry in its Events table indicating the success or failure of the operation.

3.1.10.3. Dependencies

This feature requires the completion of System Feature 9, "End election," before it can begin execution.

This feature requires Security Features 2, "Encrypt Database tables," 6, "Encrypt outbound data," 1, "Verify identity of data transmitters," and 8, "Database login."

3.1.11. System Feature 11: Detect and log errors

3.1.11.1. Purpose of Feature

This feature allows the System to detect exceptions and errors and store auditable information about them in the Database for later retrieval.

3.1.11.2. Stimulus-Response Sequence

1. While performing an operation, CardReader encounters an error or exception.
2. CardReader attempts to write the contents of the System's temporary memory (hereafter the "memory dump") to the hard disk.
3. The System's behavior depends on the type of error that is encountered.
   a. If it is a non-fatal error, CardReader records the software error code, identification of the machine on which the error occurred, date of the error, and time of the error in the "Events" Database table. CardReader also records the type of error and an error message, if provided. It then executes Step 5.
   b. If it is a fatal error that requires that the Hardware be rebooted, CardReader attempts to write the error code, date, time, error type, and error message to the hard disk. It then reboots and performs Step 4.
4. CardReader reboots the System. During bootup, CardReader reads the hard disk to determine whether the software experienced a fatal error condition the last time it shut down. Since this was the case, CardReader reads the memory dump from the hard disk.
5. Using the system state information in the memory dump, CardReader attempts to continue or restart the operation that was being performed when the error occurred.
6. CardReader resumes normal operation.
7. CardReader sends a data transmission to the precinct computer indicating that it experienced an error condition.
8. CardReader receives a data transmission from the precinct computer requesting the error information.
9. CardReader Software transmits a copy of the entry in the "Events" Database table to the precinct computer.
10. CardReader receives a data transmission from the precinct computer requesting the memory state information at the time of the error.
11. CardReader transmits a copy of the memory dump that it stored.
12. CardReader receives an acknowledgment from the precinct computer that all data were received successfully.

3.1.11.3. Dependencies

This feature requires the completion of no other System Features before it can begin execution. It does require that an error condition occur.

This feature requires Security Features 2, "Encrypt Database tables," 6, "Encrypt outbound data," 1, "Verify identity of data transmitters," and 8, "Database login."

3.1.12. System Feature 12: Conduct ballot recount

3.1.12.1. Purpose of Feature

This feature allows the CardReader Software to be configured to conduct and store the results of a machine recount of ballots. It is assumed that any recount will be conducted only on paper ballots rather than electronic votes. All InnoVote voting products produce a "paper trail" that CardReader can read.

For this operation, it is assumed that an election official has destroyed the paper ballot for any votes generated by a MyVotronic machine that are modified after a paper ballot has been printed. However, the electronic versions of such ballots would be archived in both the MyVotronic's Database and ReliaVote PE's Database, and ReliaVote PE checks recount results to ensure that no archived ballots are included.

3.1.12.2. Stimulus-Response Sequence

1. A user indicates to the System that he/she wishes to perform a recount.
2. The System determines whether the election has been declared over. Conditional: If the election is in progress, the System displays a message to that effect and exits this sequence of steps.
3. The System prompts the user if he/she wants to program the scanner to not read certain contests on the ballot. Conditional: If the user chooses to reprogram the scanner, the System performs System Feature 3, "Program the scanner," and returns to Step 4 upon successful completion of that operation.
4. The machine enters "Recount Mode," in which operations can be performed on locked ballots. (However, these operations do not modify the "Votes" table.)
5. The user inserts a ballot into the CardReader's ballot scanner.
6. The scanner scans the ballot and detects the markings on it.
7. The System scans the ballot's bar code and checks the Ballots table of the Database to determine if the ballot has been scanned. Conditional: If the ballot identification corresponding to the bar code is not present in the Database, the System adds a new entry in the Ballots table for the ballot.
8. The System compares the markings to the mappings of ballot areas and Database ballot options.
9. The System generates votes in temporary memory from marked areas of the ballot that are mapped to candidates and ballot options.
10. The System checks each contest for the presence of "overvotes."
    a. If the System detects that a contest has no more than one ballot option selected, it performs Steps 11 – 16.
    b. If the System detects a vote for more than one candidate for the same contest, it discards all the user's choices for that contest.
11. The System sets the "recounted_flag" attribute in the Database for the ballot to "TRUE."

12. The System adds the vote in RAM to the Recount_Votes Database table. It then increments the Recount_Tallies table entry for the chosen ballot option by one.
13. The System transmits a request to the precinct computer to send the data.
14. The System receives a request from the precinct computer for the data.
15. The System transmits the Recount_Votes and Recount_Tallies tables to the precinct computer.
16. The System receives an acknowledgment from the precinct computer containing the precinct computer's success or failure to add the data to its own database. Conditional: If the System receives notification of a failure, it reverts to Step 13. Conditional: If the System receives a transmission from the precinct computer to lock itself, it immediately performs System Feature 7, "Lock CardReader machine."
17. The System records an entry in its Events table indicating the success or failure of the operation.

3.1.12.3. Dependencies

This feature requires the completion of System Feature 9, "End election," before it can begin execution.

This feature requires Security Features 1, "Verify identity of data transmitters," 2, "Encrypt Database tables," 6, "Encrypt outbound data," 7, "Block all ports except two," and 8, "Database login." It is restricted by Security Features 3, "Restrict data flow to Database tables," and 4, "Limit changes to Database on Election Day."

3.1.13. System Feature 13: Clear election results

3.1.13.1. Purpose of Feature

This feature allows the System to remove all vote data and tallies from the Database after an election is over and the data are no longer needed. This operation can be performed *only* after a predetermined period of time after the day set for Election Day. It should be noted here that results cannot be deleted in part; this operation clears the entire Database table. This is an anti-fraud mechanism.

3.1.13.2. Stimulus-Response Sequence

1. The System receives one or more data transmissions from the precinct computer.
2. The System verifies the data's encryption keys.
3. The System decrypts the data and determines them to be instructions to clear the entries in the Votes, Realtime_Votes, Recount_Votes, Tallies, Recount_Tallies, and Ballots tables in its Database.
4. The System checks to ensure that enough time has elapsed and that the operation is allowed. Conditional: If enough time has not elapsed, the System displays a message indicating that the operation cannot be performed and exits this operation.
5. The System attempts to clear the affected tables in the Database.
6. The System sends a notification to the precinct computer of its success or failure to clear the data. Conditional: If the machine failed to clear the data, the notification contains the error code and error text.
7. The System records an entry in its Events table indicating the success or failure of the operation.

3.1.13.3. Dependencies

This feature requires the completion of System Feature 9, "End election," before it can begin execution.

This feature requires Security Features 2, "Encrypt Database tables," 6, "Encrypt outbound data," 1, "Verify identity of data transmitters," and 8, "Database login."

3.2. Performance Requirements.

The features described in this section are requirements that are necessary for CardReader Software to perform in a reasonable amount of time.

3.2.1. Performance Requirement 1: Modify Database quickly

Numerous features of CardReader Software require changes to be made to the Database. Any operations involving the Database must take place in an amount of time that would be unremarkable to a member of the Voter user class. The voting machine must not appear to a Voter to slow down when performing any operation.

As is described in §3.3, subsections, numerous operations require that parts of the Database employ cryptography for data protection and machine identity verification. The cryptographic algorithms used must be executed in a reasonable amount of time and be unnoticeable to a Voter.

3.2.2. Performance Requirement 2: Transmit and receive data quickly

CardReader Software has to receive, transmit, and process large amounts of data. This requires that the computer have access to a broadband Internet connection and a fast link between it and the precinct computer with which it exchanges data. CardReader itself must be able to accept the large amounts of data without slowing its processing excessively or losing any of the data.

3.3. Security Requirements.

The features described in this section are requirements that are necessary to ensure the integrity of election data generated and stored by the CardReader Software product.

3.3.1. Security Feature 1: Verify identity of data transmitters

3.3.1.1. Purpose of Feature

This feature restricts incoming traffic to the CardReader Hardware. The precinct computer (see §3.3.6) limits access to the private network, but in the event that it should fail, this security feature protects the Database from malicious operations. It requires that any data transfers processed by CardReader Software must have originated at the precinct computer or Kerberos key-management server.

3.3.1.2. Characteristics of Feature

When any data transfers are received, the system will check to ensure that they have been encrypted. The details of this system are given in reference [4], *Network Detailed Design*.

The keys will be managed by a Kerberos-compatible key management system on a server within the private network for the precinct.

Each CardReader kiosk is protected by a firewall (see *Network Detailed Design,* reference [4]). This firewall will restrict traffic to the CardReader kiosk to that originating at the precinct computer; however, CardReader software has its own backup measure and a "software firewall" in case the firewall experiences a problem and other traffic is allowed to enter the System through the CardReader data port. If CardReader Software determines that a packet was sent by the precinct's computer and intended for the machine that received it, then the packet will be processed. Otherwise, the packet is discarded.

3.3.2. Security Feature 2: Encrypt Database tables

3.3.2.1. Purpose of Feature

This feature provides for encryption of sensitive Database tables in the CardReader Software internal Database.

3.3.2.2. Characteristics of Feature

The Votes, Realtime_Votes, and Recount_Votes tables in the Database are highly sensitive and must be protected with encryption of at least 128-bit strength when not being modified. This encryption scheme can be symmetric or asymmetric; the two options are detailed in reference [2]. The keys to this encryption system must not be known or recoverable by human users.

The entire table is encrypted rather than individual entries. This means that it is not possible to add, modify, or delete entries while the table is encrypted.

When the System initiates an operation that involves one of these tables, it decrypts the affected table, performs the operation, generates a new key, and re-encrypts the table with the new key. This means that the key is changed every time an operation is performed on one of the tables. The software generates different keys for each table.

3.3.3. Security Feature 3: Restrict data flow to Database tables

3.3.3.1. Purpose of Feature

This feature restricts traffic to sensitive tables in the CardReader Software Database.

3.3.3.2. Characteristics of Feature

The Votes and Realtime_Votes tables in the Database are highly sensitive. In addition to being protected by strong encryption and secret keys, they are protected from unauthorized modification and viewing. CardReader analyzes all incoming data packets to determine if they contain instructions to modify either of these tables, and if so, the packets are discarded. ReliaVote PE should not be generating such packets (refs. [6] and [7]), and their presence could indicate that the machine running that software has been compromised. Likewise, the only machine that needs to request copies of the database is the precinct computer, and packets containing such requests are processed only if they originated on this computer. All packets that involve Database operations are “repackaged” by the CardReader software as a CardReader operation, so that the Database will recognize and execute them.

3.3.4. Security Feature 4: Limit changes to Database on Election Day

3.3.4.1. Purpose of Feature

This feature restricts user access to particular tables in the CardReader Software Database for a time period on and immediately after Election Day.

3.3.4.2. Characteristics of Feature

The Database contains numerous tables containing information about candidates, contests, and political parties. These tables can be modified by Election Officials using ReliaVote PE until 12:00 A.M. on Election Day. At this time all of the Database will be locked from modification (except for the Ballots, Votes, Realtime_Votes, Tallies, and Errors tables) for a given period of time after Election Day ends. Only the System user can modify these tables during this lockdown.

3.3.5. Security Feature 5: Private network

3.3.5.1. Purpose of Feature

This feature requires that all machines running CardReader Software must be part of a private network and have private IP addresses.

3.3.5.2. Characteristics of Feature

In a precinct, all CardReader machines and any other election equipment (hereafter "nodes") will be connected to a private network. The only node with direct access to public IP addresses will be the precinct computer. The precinct computer will filter traffic that is destined for private nodes. If a data transmission from the county's computer requests that the precinct computer initiate an operation on a CardReader machine, then the precinct computer will generate a new data transmission for the intended node, with its own IP address as the source rather than the central computer's. (It should be noted that the destination node will still discard the data packet if it contains instructions to modify a sensitive Database table.) Traffic destined for a private node originating from any other IP address is dropped.

Reference [4], *Network Detailed Design*, contains a detailed description of network security features for InnoVote products.

3.3.6. Security Feature 6: Encrypt outbound data

3.3.6.1. Purpose of Feature

This feature requires that all data transfers destined for a machine external to the CardReader Hardware must be encrypted.

3.3.6.2. Characteristics of Feature

In addition to being on a private network, any outbound data transmissions from a CardReader Hardware machine will be protected with encryption of at least 192-bit strength.

The feature utilizes the same cryptosystem described in Security Feature 1.

3.3.7. Security Feature 7: Block all ports except two

3.3.7.1. Purpose of Feature

This feature requires all data ports on the CardReader Hardware machine will be blocked from sending and receiving data transmissions except for one over which the System will exchange data with the precinct computer and another which it will use for data transfer with the Kerberos server.

3.3.7.2. Characteristics of Feature

CardReader Software will initiate connections on one data port, and this port will be used only by InnoVote software. Additionally, the client installation of Kerberos software will use a data port to communicate with the key server for the precinct network. However, a hardware firewall will be configured to disallow traffic originating from or destined for any port other than the chosen ones. The firewall must not permit any configurations that would permit incoming or outgoing traffic from ports other than these on the computer running CardReader. More information about this is present in *Network Detailed Design* [4].

3.3.8. Security Feature 8: Database login

3.3.8.1. Purpose of Feature

This feature requires that all accesses of the Database be made by a verified "user" that the database management system recognizes. This is to prevent unauthorized SQL querying.

3.3.8.2. Characteristics of Feature

The database management system will recognize the CardReader Software as a "user." The software must authenticate itself when it makes any modification to the Database. The database management system will not permit anonymous SQL querying.

3.4. System Attributes.

The attributes described in this section are, unless otherwise stated, general to the CardReader Software product rather than specific to a particular System feature.

3.4.1. Reliability

The CardReader Software must experience normal exception-free behavior at least 99.999 percent of the time. This would correspond to no more than one exception within a 24-hour period.

3.4.2. Availability

The CardReader Software will execute on a computer at all times. This product requires physical access either to a CardReader ballot scanner or a computer executing ReliaVote PE with a network connection to the computer with CardReader Software installed.

3.4.3. Security

The security requirements of CardReader are detailed in §3.3, “Security Features.”

3.4.4. Maintainability

The System must be upgradable if necessary. Any upgrades must require no changes to the existing relational schema for the Database. They must not compromise any Security Features of the software.

3.4.5. Portability

The software must execute on any computer with a hardware connection to a CardReader ballot scanning machine and the hardware device driver for the CardReader ballot scanner installed.